International Journal of Applied Engineering
Research and Development (IJAERD)
ISSN(P): 2250-1584 ISSN(E): 2278-9383
Vol. 4, Issue 5, Oct 2014, 33-42

A FUZZY-LOGIC TRAINED NEURAL NETWORK APPROACH FOR MACHINE SCHEDULE OPTIMIZATION

D. KAVITHA[1], K. G. DHARANI [2] & M. PRIYADARSINI[3]

Assistant Professors, Department of E.C.E., M.V.J College of Engineering, Bangalore, India

ABSTRACT

The competitive manufacturing global scenario today needs multi-job, multi-machine and multi criterion shop floor machine scheduling concept for higher production. Achieving key performance measures is important for the successful operation of the manufacturing systems, which are complex and dynamic in nature. To have a closer look at real-world problems, neural network concept becomes its solutions. The problem discussed in this research involves N jobs and M processors. The objective is to find a schedule which assigns N jobs to M processors in such a way that the performance measures are optimized. Dispatching rules are usually applied dynamically to schedule jobs in manufacturing systems.

Workflow balancing on a shop floor helps to remove bottlenecks present in the manufacturing system .Workflow refers to the total time during which the work centers are busy. Earlier researchers have not specified the method for jobs to be executed in parallel in order to balance the workflow to each machine. In parallel machine scheduling there are m machines to which n jobs are to be assigned based on different priority strategies. The procedure is based on the idea of workload balancing and on balancing the workload among machines. Different priority strategies are followed for the selection of jobs. 8 different strategies are considered, namely random (**RANDOM**), shortest processing time (**SPT**), longest processing time (**LPT**), most work remaining (**MWKR)**, least work remaining (**LWKR**), first come first serve (**FCFS**) and last come first serve (**LCFS**) for the selection of jobs for workflow balancing. The relative percentage of imbalance (**RPI**) is adopted among the parallel machines to evaluate the performance of these strategies in a standard manufacturing environment using neural network.

This paper discusses the application of neural network to solve a identical parallel machine scheduling. The developed neural network model predicts the optimal solutions for any set of problem instances. This paper uses the Back propagation and Delta rule based approach for training the neural network. The neural network approach is quite effective and efficient for selecting the best strategies and their optimal sequence for a given scheduling problem.

KEYWORDS: Neural Network, Fuzzy Logic, Machine Scheduling, Dispatching Rules

INTRODUCTION

Most combinatorial optimization problems such as scheduling problems are NP-hard in the strong sense, implying that it is highly unlikely that a polynomial algorithm can be designed to solve this problems. Since neural network are most effective in solving classification and prediction problems, the emerging technology of neural network may be well suited to solve this problem.

In this paper, we propose to use neural network systems to solve an identical parallel machine scheduling problems and to predict the optimal solution for any set of problem instances. Neural network have been applied extensively to a

variety of problems in various disciplines such as data mining, pattern recognition and classification. They have also been used for solving certain optimization problems.

PARALLEL MACHINE SCHEDULING

Jobs-to-machine assignment is based on several factors, such as precedence, availability, production rate, machine suitability (processing plan) and workload balancing with reference to the objective function. Researchers have addressed parallel machine scheduling problems with many variant approaches. More objective functions are considered to obtain an efficient schedule for scheduling problems. The number of machines is denoted by m and number of jobs by n. Each job is indicated by its processing time p_j where j = 1, 2, 3, ……., n.

The neural network produces an efficient schedule based on an objective such as minimisation of makespan time. The performance of the various strategies is measured by various measures relative to objective function.

There are n jobs to be processed by m machines. Any machine can process at most one job at a time. No preemption is allowed. The basic objective of workload balancing is to achieve efficient utilization of productive capacity. The concept of balancing is to evenly distribute the jobs among the machine. To evaluate the different workflow balancing strategies, the authors used the performance measure of relative percentage of imbalance (RPI).

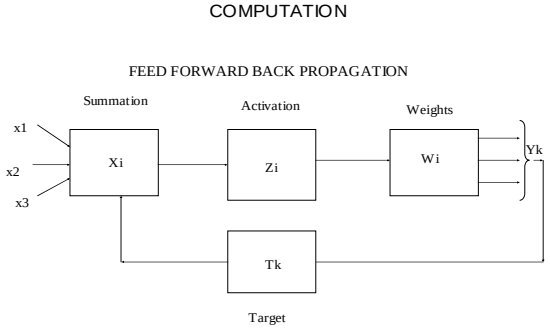
The workflow in each machine is calculated before assigning the jobs. The machine with less workflow is selected for assigning a new job from the list of jobs. The jobs are allocated to machines statically as a part of production plan. Evaluation of parallel job schedulers is based on a performance measure called workload balancing. Workload balancing aims to provide equal distribution load on parallel machines and help to reduce the total elapsed time of jobs. The aim of the scheduling process is to increase the utilization of machines and throughput.

METHODOLGY

Neural Network

NN consists of network of interconnected processing elements or neurons, arranged in layers (input, hidden, and output layers). Processing elements are connected from one layer to the next through links. These links are characterized by weights.

With the help of input, activation (or transfer) and output functions on the processing elements (PE_S) and weights on the links between processing elements, a given input is transformed to an output. Better outputs are obtained in subsequent iterations by intelligently modifying the weights (using a learning rule) on the links.



We build a NN for the given scheduling problem, with weights on links between processing elements and with input, activation and output functions on the processing elements such that the output of the final processing elements gives the makespan of a feasible solution. We then use this

Workflow Balancing in Parallel Machine Scheduling

adaptive machine-learning paradigm to find improved solutions in subsequent iterations by modifying the weights using an intelligent learning strategy. The input, activation and output functions can be designed to implement a given heuristic and to enforce the constraints of the problem. Various learning strategies can be devised. This approach thus offers flexibility of using any established heuristic with any learning strategy.

Fuzzy Scheduling

To solve the scheduling problem with a fuzzy – based system the following sub – problems need to be addressed .Modeling and transformation of scheduling data into a knowledge representation that can be handled by a fuzzy controller (fuzzification) Processing of the fuzzy scheduling knowledge towards a decision by means of given rules and integration of fuzzy arithmetic to deal with imprecise or vague data and Transformation of the fuzzy scheduling decision into crisp scheduling data (defuzzification)

Knowledge traditionally represented by IF – THEN rules is implicitly and distributive represented by connection weights and local processing units in the network. Each unit is responsible for many rules with different activation strengths. In other words, the function of each IF – THEN rule is undertaken by many units. It seems that a disturbed network inherently possesses some fuzziness which may come from the distributiveness of the network although the term fuzziness here is used more technically and less conceptually . On the other hand , instead of paying primary attention to the functional aspect, the localized method is concerned mainly with the structural aspect , that is , to seek a close structural equivalence between the logic - based algorithm and the neural network structure . In this regard , each IF – THEN rule can be represented by only one unit in the network with associated weight vectors accounting for the IF part and THEN part . Thus knowledge is represented ever, the computing (reasoning) process for deriving an appropriate action in response to an input may be competitive or co –operative depending s on the corresponding reasoning algorithms .

Training Algorithm

Step 0: Initialize weight, (set of small random values).

Step1: While stopping condition are false , do steps 2-9.

Step2: For each training pair, do steps 3-8.

Feed forward:

Step 3: Each input unit (x_i, i=1, 2 ,….p) receives input signal xi and broadcast this signal to all units in the layer above (the hidden units)

Step 4: Each hidden unit (z_j, j=1, 2 ,…..p) sums its weighted input signal

$$Z_{,inj} = V_{Oj} + \Sigma X_i V_{ij},$$

Applies its activation function to compute its output signal

$$Z_j = f(Z_{,inj}),$$

And sends this signal to all units in the layers above (output units)

Step 5 : Each output unit (Y_k, k=1,2,3,….m) sums its weighted input signals.

$$Y_{ink} = W_{ok} + \Sigma Z_j W_{jj}$$

And applies its activation function to compute its output signal

$$Y_k = f(Y_{ink})$$

Back propagation of error :

Step 6 : Each output unit (Y_k, k=1,2,3,….m) receives a target pattern corresponding to the input training pattern, computes its error information term.

$$\Delta k = (t_k - y_k)\, f(Y_{ink}).$$

Calculates its weights correction sum(used to update work later)

$$\Delta W_{jk} = \alpha S\, KZ_j$$

Calculates its bias correction sum(used to update work later)

$$\Delta W_{ok} = \alpha Sk$$

And sends S_K units in the layer below.

Step 7 : Each hidden unit (Z_j = j=1,2,..p) sums its delta inputs (from units in the layer above)

$$S_{inj} = \Sigma S_k W_{jk}$$

Update weights and biases:

Step 8 : Each output unit (Y_k ,k =1,2,..m) update its bias and weights (j=0,1,2,…p)

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk}$$

Each hidden unit (Z_j ,j=1,2,….p) updates its bias and weights (I = 0,1,2….n)

$$V_{ij}(new) = V_{ij}(old) + \Delta V_{ij}$$

Step 9 : Test for stopping condition.

TRAINED DATA'S USING NN

From m=(10 to 100) data's , optimal solution were recorded. These optimal solutions (m = 10 to 70) have been used to train the Neural Network and the optimal solutions (m = 80 to 100) have been used to test the NN model.

TRAINING DATA

Table 1

S.No	No.of.Jobs	Weightage to Earliness	Weightage to Tardiness	Earliness	Tardiness
1	10	0	1	8151	0
2	10	0.1	0.9	6977	0
3	10	0.2	0.8	6716	0
4	10	0.3	0.7	6647	0
5	10	0.4	0.6	6992	0
6	10	0.5	0.5	7022	0
7	10	0.6	0.4	7022	0
8	10	0.7	0.3	6716	0
9	10	0.8	0.2	6775	0
10	10	0.9	0.1	6908	0
11	10	1	0	6551	0
12	20	0	1	10731	2533
13	20	0.1	0.9	8613	2341
14	20	0.2	0.8	10768	2336
15	20	0.3	0.7	6552	2930
16	20	0.4	0.6	7580	3119
17	20	0.5	0.5	9654	1528
18	20	0.6	0.4	6899	3577
19	20	0.7	0.3	8291	1930
20	20	0.8	0.2	6902	3267
21	20	0.9	0.1	6584	6038
22	20	1	0	6092	5139
23	30	0	1	7128	54702
24	30	0.1	0.9	4587	62483
25	30	0.2	0.8	6423	61885
26	30	0.3	0.7	432	75434
27	30	0.4	0.6	2867	64501
28	30	0.5	0.5	931	104630
29	30	0.6	0.4	1902	72228
30	30	0.7	0.3	680	105805
31	30	0.8	0.2	1068	94037
32	30	0.9	0.1	842	102217
33	30	1	0	352	122213
34	40	0	1	8549	188133
35	40	0.1	0.9	3915	199685
36	40	0.2	0.8	4432	179940
37	40	0.3	0.7	0	242560
38	40	0.4	0.6	649	249876
39	40	0.5	0.5	640	267108
40	40	0.6	0.4	312	267306
41	40	0.7	0.3	195	264570
42	40	0.8	0.2	0	277364
43	40	0.9	0.1	0	283797
44	40	1	0	159	328004
45	50	0	1	8065	423624
46	50	0.1	0.9	406	484361
47	50	0.2	0.8	497	452975
48	50	0.3	0.7	1082	481388
49	50	0.4	0.6	594	486370
50	50	0.5	0.5	206	499521
51	50	0.6	0.4	352	531710
52	50	0.7	0.3	98	546479
53	50	0.8	0.2	0	590396
54	50	0.9	0.1	0	683949
55	50	1	0	0	480380
56	60	0	1	6577	850691
57	60	0.1	0.9	1487	893018
58	60	0.2	0.8	732	948028
59	60	0.3	0.7	80	992130
60	60	0.4	0.6	0	1041992
61	60	0.5	0.5	108	926131
62	60	0.6	0.4	108	936102
63	60	0.7	0.3	0	1084021
64	60	0.8	0.2	0	1073312

Table 1: Contd.,

65	60	0.9	0.1	0	926028
66	60	1	0	0	1079989
67	70	0	1	4245	1280656
68	70	0.1	0.9	538	1384698
69	70	0.2	0.8	992	1321050
70	70	0.3	0.7	188	1513371
71	70	0.4	0.6	188	1455394
72	70	0.5	0.5	0	1556575
73	70	0.6	0.4	0	1530283
74	70	0.7	0.3	0	1515503
75	70	0.8	0.2	0	1509272
76	70	0.9	0.1	0	1555171
77	70	1	0	0	1924361

TESTING DATA

Table 2

S.No	No. of. Jobs	Weightage to Earliness	Weightage to Tardiness	Earliness	Tardiness
1	80	0	1	10677	1835640
2	80	0.1	0.9	370	1921364
3	80	0.2	0.8	0	1925112
4	80	0.3	0.7	80	2006730
5	80	0.4	0.6	98	2020057
6	80	0.5	0.5	0	1987950
7	80	0.6	0.4	98	1958594
8	80	0.7	0.3	0	2080021
9	80	0.8	0.2	0	1943192
10	80	0.9	0.1	0	2078870
11	80	1	0	0	2437276
12	90	0	1	6093	2399014
13	90	0.1	0.9	188	2525420
14	90	0.2	0.8	870	2209929
15	90	0.3	0.7	0	2447034
16	90	0.4	0.6	0	2407299
17	90	0.5	0.5	0	2537169
18	90	0.6	0.4	0	2639539
19	90	0.7	0.3	0	2585193
20	90	0.8	0.2	0	2744345
21	90	0.9	0.1	0	2403858
22	90	1	0	0	3080468
23	100	0	1	0	2877253
24	100	0.1	0.9	0	2890696
25	100	0.2	0.8	0	2914180
26	100	0.3	0.7	262	3013525
27	100	0.4	0.6	0	3093637
28	100	0.5	0.5	150	2964124
29	100	0.6	0.4	0	3031905
30	100	0.7	0.3	0	2890696
31	100	0.8	0.2	0	3196729
32	100	0.9	0.1	0	2877253
33	100	1	0	0	3309080

RELATIVE PERCENTAGE OF IMBALANCE

The relative percentage of imbalances (RPI) in workload of all the machines is adopted to compare the performance of the scheduling strategies . This indicates the percentage of deviation of workloads o machines from the upper bound of maximum work loads on machines from the bound of maximum work load. It is expressed as

Relative percentage of imbalance (RPI) or

$$\text{Percentage of deviation from upper bound} = \frac{(W_{max} - W_i)}{W_{max}} \times 100$$

The average RPI values are calculated using Excel using the out put of the simulation model . The out put of the simulation model shows the work allotments to the machines in terms of job processing times .The 5 scheduling strategies produce 5 different work allotment schedules .The maximum workload on the W max machine is identified and the percentage of deviation form this maximum workload for all the machines is calculated individually for each of the 5 rules. The workload is balanced perfectly if the percentage of imbalance among the machines is zero.

RESULTS AND DISCUSSIONS

The RPI values make span & mean value flow time job sizes using Neuro - fuzzy approach are compared with HM- Algorithm . The result shows that Neuro - Fuzzy approach gives the results , when compared with the Algorithms . The performance of various strategies using Neural network is discussed below. While company the strategies , the SPT rule shows lesser percentage of imbalance and Mean Flow Time for all the M Machine problems. This SPT gives least values of RPI below 1 % for job sizes of n = 4 in 2 machines, n=6 in 3 machines , n = 8 in four machines , n=10 in five machines problems . This indicates that when the number of machines for scheduling in increased , the lowest RPI values fall below s 1 % for the increased job sizes . A shift of this RPI values on the right hand side is observed in with an increase in n . All the five strategies start with higher RPI values for smaller range of n . The RPI values decreases for machine range of n & a higher of n . The SPT rule seems to produce good schedules with a lesser percentage of imbalances for all the M machines problems . After reaching the minimum of less than 1 % in all 5 cases it forms a straight line & involves horizontally along the X – axis.

These RPI values of SPT is less than the RPI values produced by other four rules . It is evident that the LCFs rule does not produce a good schedule with respect to workloads when the jobs size is lower on m machines .The RANDOM lines lie between LCFS & LPT, lines LCFS & LPT lies between SPT & FCFS lines in all the cases , all the five rules shows a higher percentage of imbalance in work loads on machines with a smaller s . The research indicates that SPT strategy produces balanced workloads in parallel machine scheduling irrespective of job size & no of machines.

Table 3: RPI Results for Various Strategies

Jobs	Random	LCFS	FCFS	LPT	SPT
4	20.64	19.34	17.32	13.13	10.16
6	18.11	17.94	15.12	12.13	9.54
8	16.00	15	13.89	11.68	8.76
10	15.43	14.87	11	8.45	5.4
15	13	11.9	8.76	6.93	4.2
20	11.3	9.8	7.2	5.4	3.2

Table 3: Contd.,					
25	9.8	5.3	4.19	4	2.9
30	7	3.7	3.5	3.6	9
35	6.4	3.	3.5	3.0	1.45
40	2.47	1.2	0.95	0.67	0.21

A comparison of Neuro-fuzzy results with Hm – Algorithm

NN – Fuzzy

Mean Flow time = 0.25

RPI = 0.0001

Hm – Algorithm

Mean Flow time = 13

RPI = 4.823

GRAPH

An comparative results of heuristic approach vs neural network

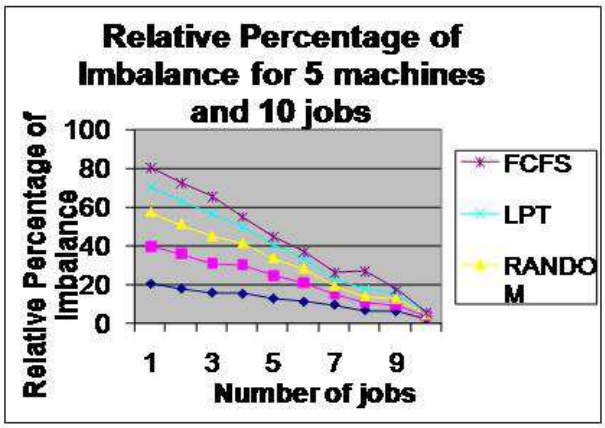


Figure 1: RPI for 5 Machines and 10 Jobs

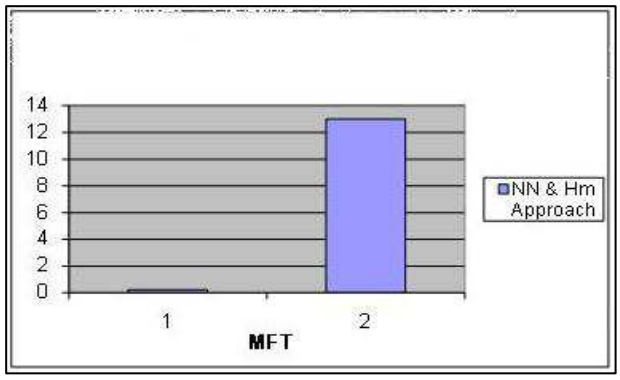


Figure 2: Neural Network Vs Hm Approach for Mean Flow Time (MFT)

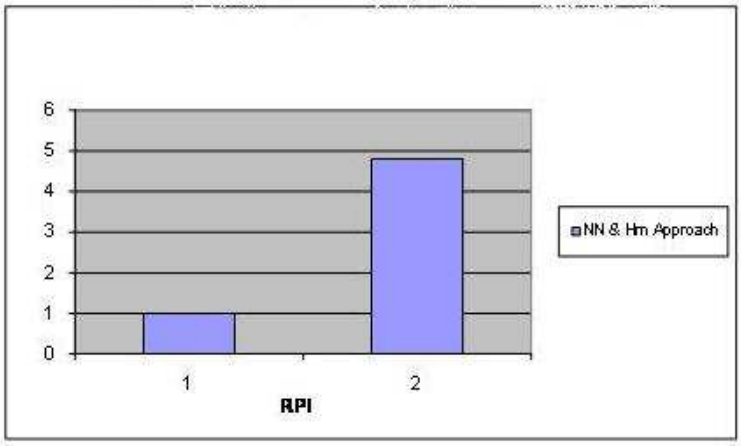


Figure 3: Neural Network Vs Hm Approach for RPI

CONCLUSIONS

In this paper we propose a neural network framework for solving the parallel machine-scheduling problem. The problem considered is that of optimizing the makespan, earliness and tardiness measures for scheduling n tasks on m identical machines. By varying the weights of makeaspan, earliness and tardiness, the optimal performance measures are found, and the best-weighted combination is identified.

These optimal solutions are used for training the neural network. Through weights assigned to the links between tasks and machines, and by adjusting the weights using an appropriate learning strategy, a significantly improved schedule is found using ANN. It has been observed that the trained NN models perform better when compared to heuristic methods.

REFERENCES

1. Sin-seok Seo; Joon-Myung Kang; Agoulmine N (2011), “A fuzzy-based adaptive scheduling technique for IEEE 802.16 networks” IEEE conference.
2. Li Yun; Kong Xian-ming (2011), “Global optimization scheduling based on fuzzy neural network” volume -1
3. Xiying Ding; Zhe Wang (2009), “A Fuzzy Neutral Network Controller Based on Optimized Genetic Algorithm for UC Rolling Mill” vol – 2
4. Jatinder N.D.Gupta (2010), “Selecting Scheduling Heuristics Using Neural Network”, INFORMS Journal on Computing, VOL-12, No: 2.
5. Harvey B.Newman, Josif C.Legrand (2010), “A Self- Organizing Neural Network for job scheduling in Distributed Systems”.
6. Anurag Agarwal, Hasan Pirkul, Varghese S.Jacob (2010). “Augmented Neural Networks for task Scheduling”, European Journal of Operational Research.
7. Panneerselvam, “Production and Operation Management”.
8. Kenneth R.Baker, “Introduction to Sequencing and Scheduling”.
9. S.Rajasekaran and G.A.Vijayalakshmi Pai, “Neural Network, Fuzzy Logic and Genetic Algorithm Synthesis and Applications”.